

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

5 **Listing of Claims:**

Claim 1 (currently amended): A method for program debugging, the method comprising:

(a) setting a plurality of breakpoints corresponding to a plurality of events in an implementation under test, each event being a test executed to a peripheral device and taking a general processing path when the peripheral device is working well or an error processing path when the peripheral device is out of order;

(b) executing the implementation under test for outputting a diagnosis code of a breakpoint;

(c) resetting a parameter of the event corresponding to the diagnosis code; and

(d) executing the event according to the reset parameter for making the event undergo ~~an error handler~~ the error processing path.

Claim 2 (currently amended): The method of claim 1 further comprising: after step (d), repeating steps (b) to (d) for making the implementation under test make all events undergo the ~~error handler~~ the error processing path.

Claim 3 (original): The method of claim 1 wherein the breakpoints are set ahead of program codes of the corresponding events.

Claim 4 (original): The method of claim 1 wherein the breakpoints are set after program codes of the corresponding events.

Claim 5 (cancelled)

Claim 6 (currently amended): The method of claim 1 wherein the ~~error handler is~~ error processing path produces an audible tone.

5 Claim 7 (currently amended): The method of claim 1 wherein the ~~error handler is~~ error processing path causes a system reset.

Claim 8 (currently amended): The method of claim 1 wherein the ~~error handler is~~ error processing path causes a system execution interrupt.

10

Claim 9 (cancelled)

Claim 10 (new): A method for program debugging, the method comprising:

generating a script file comprising a user assigned diagnosis code, an first parameter,

15

and a second parameter corresponding to an event in an implementation under test, the event being a test executed to a peripheral device and taking a general processing path when the peripheral device is working well or an error processing path when the peripheral device is out of order;

setting a breakpoint corresponding to the event in the implementation under test;

20

executing the implementation under test using a parameter equal to the

corresponding first parameter for outputting a diagnosis code of the breakpoint; the script file resetting the parameter to be equal to the second parameter when the diagnosis code answers to the user assigned diagnosis code; and executing the event according to the second parameter.

25

Claim 11 (new): The method of claim 10 further comprising executing the event according to the general processing path when the first parameter is the same as the second parameter.

Claim 12 (new): The method of claim 10 further comprising executing the event according to the error processing path when the first parameter is different from the second parameter.

5

Claim 13 (new): The method of claim 12 wherein the error processing path comprising a generic event error handler and a critical event error handler, each error handler being respectively executed according to different second parameters.

10 Claim 14 (new): The method of claim 13 wherein the generic event error handler displays error messages or writes errors into a file or both.

Claim 15 (new) The method of claim 13 wherein the critical event error handler and produces at least one of an audible tone, a system reset, and a stop execution command.

15

Claim 16 (new) The method of claim 13 further comprising executing the implementation under test until the end of the program after undergoing the error processing path.

20